METHOD AND APPARATUS FOR MONITORING AND UPDATING SYSTEM SOFTWARE

FIELD OF THE INVENTION

This invention pertains to monitoring software for a variety of conditions such as internal performance characteristics, liability warnings, programmatic errors and the general health of the computer system.

BACKGROUND OF THE INVENTION

No matter the computer program, it is inevitable that there will be some bugs (that is, coding errors that cause the program to behave differently what is expected). Production environments represent a number of variables that are difficult to reproduce in testing environments. As such, applications with thousands of interfaces can fail under a variety of changing variables.

Because human intervention is required to maintain these applications, certain tasks must be completed by operations on a timely basis. Failure to operate and maintain the system within the published guidelines for the application will result in a number of unacceptable issues. These include, but are not limited to the following: inaccurate reporting of revenues; increased risks associated with liability; increased risks with system availability; and increased costs due to additional manpower correction activities.

Customers want to know that their mission critical system is performing at peak levels of performance. They want to know when an area of the system is failing. They need to feel confident that the system and its integration with operations are running smoothly. Not knowing the health of the internal components of the system can create a false sense of security.

Another thing software companies sometimes do to eliminate defects is to find out about defects from customers. For a long time, customers had to make contact with the software companies (either by telephone or by e-mail) and let the software companies know about the bugs. More recently, as exemplified by Microsoft® Windows® XP, the operating system offers to send an error report to the software company when a program crashes. That

way, the software company is informed about serious errors. (Microsoft and Windows are registered trademarks of Microsoft Corporation in the United States and other countries.)

Some third party products that monitor the operation of systems from the outside exist. For example, Netcool, by Micromuse, collects information from APIs, log files, and other utilities, and forwards this information to a server for filtering. Patrol, by BMC Software, offers remote monitoring and full-application management. But both of these products are external to the applications being monitored. These products focus primarily on external environments surrounding the application. They cannot detect the internal health of the application itself and thus their reporting value is limited in scope.

A need remains for a way to proactively detect application problems and software defects through monitoring internal application performance beyond that associated with the prior art.

SUMMARY OF THE INVENTION

The invention is an apparatus, system, and method for monitoring computers. A series of probes residing on a customer's computer determines values for metrics and sends these values to a monitoring apparatus. The monitoring apparatus determines if the values for the metrics are acceptable. If the values for the metrics are not acceptable, then an alert is displayed so that a corrective measures can be initiated.

The foregoing and other features, objects, and advantages of the invention will become more readily apparent from the following detailed description, which proceeds with reference to the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 shows a central server with a monitoring apparatus, communicating with probes on customer computers, according to an embodiment of the invention.

FIG. 2 shows details of the monitoring apparatus of FIG. 1.

FIG. 3 shows details of the probes of FIG. 1.

FIG. 4 show the probes of FIG. 1 communicating with the monitoring apparatus of FIG. 1.

FIG. 5 shows a flowchart of the procedure for using the probes of FIG. 1.

FIGs. 6A-6C show a flowchart of the procedure for using the monitoring apparatus of FIG. 1.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

FIG. 1 shows a central server with a monitoring apparatus, communicating with probes on customer computers, according to an embodiment of the invention. In FIG. 1, server 105 is a central server. Server 105 is operated by the company distributing software products to its customers. Customers operate, for example, computers 110, 115, and 120. Although a person skilled in the art will recognize that there can be more or fewer than three customers, and that each customer can have more than one computer.

Installed on computers 110, 115, and 120 are probes 125, 130, and 135. Probes 125, 130, and 135 are responsible for determining the values associated with various metrics on computers 110, 115, and 120 respectively, and transmitting these values back to server 105. The details of probes 125, 130, and 135 are discussed further with reference to FIGs. 3-4 below.

Server 105 includes monitoring apparatus 140. Monitoring apparatus 140 receives information from probes 125, 130, and 135, and determines whether the data received from the probes represent acceptable values. If the values are acceptable, then monitoring apparatus 140 logs the values. Otherwise, monitoring apparatus displays 140 an alert, indicating the unacceptable value. The details of monitoring apparatus 140 are discussed further with reference to FIG. 2 below.

Connecting server 105 with computers 110, 115, and 120 is network 145. Network 145 can be any variety of network including, among others, a local area network (LAN), a wide area network (WAN), a global network (such as the Internet), and a wireless network (for example, using Bluetooth or any of the IEEE 802.11 standards). In addition, a person skilled in the art will recognize that different networks can be used to connect server 105 with different computers. For example, server 105 might be connected to computer 110 using one network, and to computers 115 and 120 using a second network.

FIG. 2 shows details of the monitoring apparatus of FIG. 1. Monitoring apparatus 140 includes four components: receiver 205, tester 210, alerter 215, and log 220. Receiver 205 is responsible for receiving a message from a probe and parsing the message for the necessary information. Tester 210 then tests the value (or values) retrieved from the message received by receiver 205 to determine if the value is acceptable. Alerter 215 displays an alert is the value retrieved from the message is not acceptable. And log 220 includes entries, like

entry 225, which reflect the received message, its values, and/or whether the value is acceptable.

To determine whether a value is acceptable, monitoring apparatus 140 uses database 230. Database 230 includes filters, such as filters 235, 240, and 245, which identify what values are considered acceptable. Different filters exist for different metrics. For example, filter 235 is a filter for the number of transactions occurring at a given location, whereas filter 245 is a filter for the number of open days experienced at a location.

Some filters, such as filter 245, can be used for all casino locations. But other metrics, such as the number of transactions, can vary from one location to another. To account for differing interpretations of acceptable values, different filters can be set up for a single metric, each filter identifying acceptable values for a different casino location. Thus, while filters 235 and 240 both represent acceptable values for the transactions metric, they represent acceptable values for different casinos.

Although a different filter can be set up for each different site for a given metric, the amount of variation in acceptable values might be limited. Where two or more sites agree on what constitutes an acceptable value for a given metric, there is no need for each site to have a separate filter. Thus, while FIG. 2 shows filters 235 and 240 being used for individual sites, a person skilled in the art will recognize that a single filter can be used for some (but not necessarily all) sites.

To select the appropriate filter, monitoring apparatus 140 uses selector 250. Selector 250 uses information from the message to select the appropriate metric. Selector 250 determines the metric represented in the message and, if necessary, the site from which the metric was measured. Selector 250 then uses these pieces of information to find the appropriate filter in database 230, so that tester 210 can determine if the value is acceptable.

FIG. 3 shows details of the probes of FIG. 1. In FIG. 3, probe 125 includes sensors 305, 310, and 315. Each sensor operates to determine values for different metrics for computer 110. For example, sensor 305 determines the number of transactions that occur in a given day in database 320 on computer 110, sensor 310 determines the number of open days at the site, and sensor 315 determines the number of fixes applied to software 325 on computer 110. A person skilled in the art will recognize that although three sensors are shown within probe 125, there can be fewer or more sensors in a given probe. In addition, there can be more than one probe for a given computer, each with the same or differing numbers of sensors.

Because sensor measurements are taken more than once, each of sensors 305, 310, and 315 includes a corresponding timer 330, 335, and 340. The timers ensure that the sensors take measurements according to regular schedules. Each timer can be set to measure a metric using different intervals. But a person skilled in the art will recognize that, for sensors measuring metrics according to consistent schedules, a single timer can be used for more than one sensor.

Additionally, sensors can trigger on two different mechanisms. They can be triggered on a timer or they can be triggered by an impromptu event. The latter is utilized to signal immediate attention to a critical event that has just taken place.

Finally, probe 125 includes message generator 345. Message generator 345 takes the measurements from the various sensors 305, 310, and 315, and assembles a message from the measurements. The message is then sent to the central server (not shown in FIG. 3). Message generator 345 can generate a single message for several metric measurements, or message generator 345 can generate a separate message for each metric measurement.

FIG. 4 show the probes of FIG. 1 communicating with the monitoring apparatus of FIG. 1. In FIG. 4, message generator 345 is shown generating message 405. Message 405 is shown in greater detail in blow-up 410. The message is dated August 7, 2003, and is from site 1 (which includes computer 110). Blow-up 410 shows two metric measurements. The site has measured 500,000 transactions, and has five open days. There can also be other metrics included in the message.

Once message 405 is generated, it is delivered to e-mail server 415. E-mail server is responsible for starting message 405 along its journey to receiver 205 in the central server. Although shown as a component of computer 110, a person skilled in the art will recognize that e-mail server 415 can be part of a separate computer, distinct from computer 110, or can be a dedicated e-mail server. A typical implementation would most likely utilize the customer's existing e-mail implementation. This will provide a number of benefits including a cost savings through the elimination of a second server along with cost avoidance of supporting and maintaining the additional hardware.

FIG. 5 shows a flowchart of the procedure for using the probes of FIG. 1. At step 505, the probe accesses a value for a metric. The value can be accessed from a database or from software. As shown by arrow 510, step 505 can be repeated as often as necessary, to access values for multiple metrics. At step 515, a message is generated. At step 520, the message includes the value for the metric accessed in step 505. At step 525, the site is

included in the message, so that the central server knows from where the message originated. At step 530, the message is delivered to the e-mail server, and at step 535, the message is sent to the monitoring apparatus by the e-mail server.

FIGs. 6A-6C show a flowchart of the procedure for using the monitoring apparatus of FIG. 1. In FIG. 6A, at step 605, the monitoring apparatus receives a message from a probe. At step 610, the metric is determined from the message. At step 615, a value for the metric is determined. At step 620, a site for the probe is determined.

At step 625 (FIG. 6B), the monitoring apparatus determines if the metric is site-specific. If the metric is site specific, then at step 630 the monitoring apparatus determines an acceptable value or range of values for the metric/site combination. Otherwise, at step 635, the monitoring apparatus determines an acceptable value or range of values for the metric, without regard to the site of the probe. Either way, at step 640, the system compares the value from the message with the acceptable value/range.

At step 645 (FIG. 6C), the monitoring apparatus determines if the value for the metric is acceptable. If the value is acceptable, then at step 650 the monitoring apparatus logs the value for the metric and the site from which the value was received. Otherwise, at step 655, the monitoring apparatus displays an alert, letting someone know about a potential problem.

As shown in FIG. 6A, certain steps can be omitted or repeated. For example, since a single message can include values for multiple metrics, steps 610 and 615 can be repeated. Also, if the metrics are not site-specific, step 620 can be omitted (although typically the site is transmitted as part of the message, even if the metric is not site-specific). Finally, as shown on FIG. 6C, if the value for the metric is acceptable, the value does not need to be logged, although, again, typically the value is logged.

A person skilled in the art will recognize that an embodiment of the invention described above can be implemented using a computer. In that case, the method is embodied as instructions that make up a program. The program may be stored on computer-readable media, such as floppy disks, optical discs (such as compact discs), or fixed disks (such as hard drives), and can be resident in memory, such as random access memory (RAM), read-only memory (ROM), firmware, or flash RAM memory. The program as software can then be executed on a computer to implement the method. The program, or portions of its execution, can be distributed over multiple computers in a network.

Having illustrated and described the principles of the invention in a preferred embodiment thereof, it should be readily apparent to those skilled in the art that the invention

can be modified in arrangement and detail without departing from such principles. All modifications coming within the spirit and scope of the accompanying claims are claimed.